StratusLab

Enhancing Grid Infrastructures with
Virtualization and Cloud Technologies

Cloud-like Management of Grid Sites 1.0 Software

Deliverable D6.2 (V1.1)
13 May 2011

Abstract

This document presents the features implemented for the automatic deployment and dynamic provision of grid services, and for the scalable cloud-like management of grid site resources. These features, developed largely in Work Package 6 (WP6) are integrated into the StratusLab Toolkit by Work Package 4 (WP4). They involve cloud-like APIs, a service definition language, contextualization, scalable cloud frameworks, monitoring and accounting solutions. Some functionalities developed include TCloud and OCCI implementations, a library to process OVF, the Claudia framework and integration with Ganglia monitoring information.

CAPACITIES

StratusLab is co-funded by the
European Community's Seventh
Framework Programme (Capacities)
Grant Agreement INFSO-RI-261552.

Contributors

Name	Partner	Sections
Muñoz Frutos, Henar	TID	2, 3, 4, 5
Huedo, Eduardo	UCM	3, 4
Montero, Rubén S.	UCM	3, 4
Floros, Vangelis	GRNET	3, 4
Loomis, Charles	LAL	General and English Revision

Document History

Version	Date	Comment
0.1	06 Apr. 2011	Initial version for comment.
0.2	14 Apr. 2011	Introduction
0.3	25 Apr. 2011	TID implemented features
0.4	27 Apr. 2011	UCM implemented features
0.5	28 Apr. 2011	GRNET implemented features
0.6	28 Apr. 2011	Executive Summary and Conclusions
0.7	3 May 2011	Peer revision
0.8	7 May 2011	Contribution towards the peer revision comments
0.9	10 May 2011	Further English and typographic corrections
1.0	12 May 2011	Simple corrections from external reviewer
1.1	13 May 2011	Final version

Contents

List of Figures

List of Tables

1 Executive Summary

Grid infrastructures are typically static, with limited flexibility for changing application parameters: OS, middleware and resources in general. By introducing cloud management capabilities, grids can become dynamic. Adding standard tools, such as virtual machines (VMs), resources can be repurposed on demand to meet the requirements of high priority applications. The cloud platform controls which application images should be running and when. This means dynamic application stacks on top of the available infrastructure, such that any physical resource can be timely repurposed on demand for additional capacity.

The Joint Research Activity (JRA), carried out in WP6, develops advanced technology/features for deployment on existing Cloud infrastructures through automatic deployment and dynamic provision of grid services as well as scalable cloud-like management of grid site resources. More specifically, the objectives to be accomplished can be expressed as: i) the extension of currently available service-level open-source elasticity frameworks on top of cloud infrastructures, ii) the invention of new techniques for the efficient management of virtualized resources for grid services and iii) the inclusion of novel resource provisioning models based on cloud-like interfaces.

Development efforts have been planned to close the gaps identified in the deliverable D4.1 *Reference Architecture for StratusLab Toolkit 1.0*. Different solutions were considered in D6.1 *Cloud-like Management of Grid Sites 1.0 Design Report*. Concretely, TCloud [5] and OCCI [2] APIs were chosen as the cloud-like APIs for accessing the Service Manager and Virtual Machine Manager, respectively. The Open Virtualization Format (OVF) [1] was selected as the service definition language and also used for contextualization. The framework Claudia [6] will handle service management and scalability, using service providers to dynamically control the grid service provisioning and scalability in an IaaS cloud. Finally, the Ganglia framework can be used for providing monitoring information to OpenNebula and Claudia.

After the architecture defined in D6.1 *Cloud-like Management of Grid Sites 1.0 Design Report* [4], some development work has been carried out to implement the solutions defined. These development functionalities have been categorized according to the gaps they close: Cloud-like APIs, Virtual Appliance Language, Virtual Machine Manager, Service Management, Service Scalability, Monitoring, and Accounting.

These implemented features, developed in WP6, have been integrated into the StratusLab distribution with help of WP4. It means that the implemented features are stored in the StratusLab git repository, accessible by the RPM package repository and testing via jobs in hudson.

This document presents the features implemented for the automatic deployment and dynamic provision of grid services, and for the scalable cloud-like management of grid site resources. It describes the implemented features, where the software can be obtained, installation documentation, configuration instructions, and end-user documentation. In addition, it lists the functionalities which are in the roadmap to be implemented in near future.

2 Introduction

WP6 objectives include the identification, design and implementation of the technology and features for grid deployment on existing cloud infrastructures. These features move towards an automatic deployment and dynamic provision of grid services as well as scalable cloud-like management of grid site resources. To achieve these, the main work done in this work package involves:

- On top of the current virtual machine manager, the introduction of a service manager, which is able to control and configure grid services as a whole, considering the service concept as a set of virtual machines and the interconnecting network. The first example for that involves the deployment and configuration of a gLite-based grid site.
- The introduction of scalability mechanism at the service-level (as part of the service manager) to scale up and down grid services components according to some Key Performance Indicators (KPIs) and hardware usage.
- The inclusion of cloud-like interfaces based on standards to increase the interoperability in cloud service management.
- Monitoring systems at the physical, virtual hardware and grid service layers for regular administration tasks, accounting purposes and for triggering scalability mechanisms.
- Accounting information to keep track of resource usage in order to apply site usage policies for fair share and potential resource charging.

According to the above objectives and taking into account a set of gaps identified in in the deliverable D4.1 *Reference Architecture for StratusLab Toolkit 1.0*, the document D6.1 *Cloud-like Management of Grid Sites 1.0 Design Report* presented a set of solutions to get the automatic deployment and dynamic provision of grid services as well as cloud-like management of grid site resources.

To this end, TCloud [5] and OCCI [2] APIs were chosen as cloud-like APIs for accessing to the Service Manager and Virtual Machine Manager, which allow i) grid users to manage their grid services by the service manager and ii) the service manager to access to the virtual machine manager to deploy the grid virtual machines into physical resources. The Open Virtualization Format (OVF) [1] was

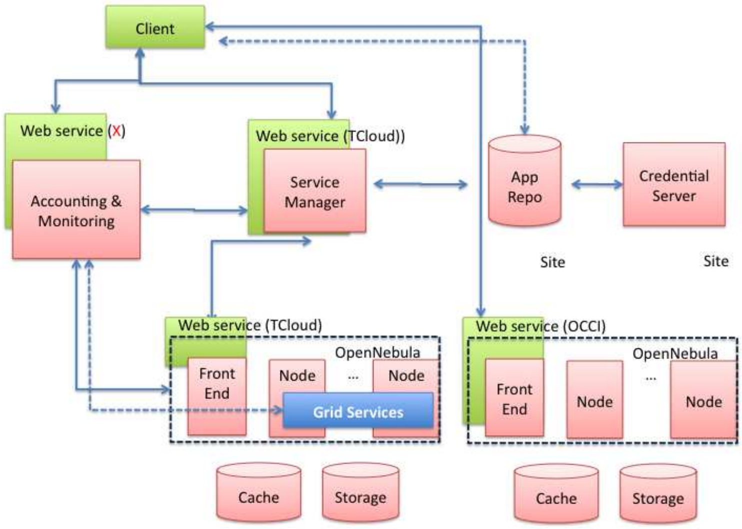

Figure 2.1: *Final StratusLab Architecture*

selected as the service definition language which provides a standard way to define the grid service requirements in terms of virtual hardware resources and grid software configuration issues. Contextualization required to configure correctly the different virtual machine according to the information described in the service definition language is also formalized in OVF. The framework Claudia [6] is used for handling service management and scalability, allowing grid users to dynamically control the grid service provisioning and scalability in an IaaS cloud. Finally, monitoring and accounting are important functionalities for checking the grid service status and tracking the resource usage.

Using the solutions identified in D6.1, an update of the architecture was performed. This updated architecture can be seen in Figure 2.1, where OpenNebula acts as the virtual machine manager. On top of it, and in order to leverage the service abstraction for the end-user, there is a Service Manager (SM). The SM is responsible for managing the overall service (defined by OVF) and providing service scalability according to service providers' requirements. The access to OpenNebula and service manager is through the OCCI and TCloud APIs. Regarding monitoring, Ganglia has been identified as a system that can provide the physical and virtual hardware monitoring information, which is used to feed OpenNebula and Claudia. Accounting systems are still in an analysis status and will be included in the coming months.

2.1 Summary of WP6 Implemented Features

In order to provide the updated architecture defined in Figure 2.1, some development has been done to begin to close the previously-identified gaps along the lines of solutions specified in D6.1 [4]. Table 2.1 provides a summary of the features to be implemented, organized according to the topics described in D6.1 [4] which close the the gaps identified in D4.1 [3].

2.2 Organization of Following Chapters

The document is organized as follows: Chapter 2 provides a summary of the features to be implemented in WP6 according to the design done in D6.1. The implemented functionalities are described in Chapter 3 while the missing ones are discussed in Chapter 4. Finally, the Chapter 5 provides some conclusions.

Table 2.1: *Status of WP6 Features*

Topic	Feature	Implemented?
Cloud like API	TCloud API	Y
	OCCI API	Y
	Authentication and authorization in OpenNebula	Y
	Authentication and authorization in TCloud	N
	Support for groups and roles	N
Virtual Appliance Language	OVF as the VApp language (ovf-manager library)	Y
Virtual Machine Manager	Improved fault tolerance	Y
	Cluster support	Y
	Graphical User interface	N
	Improved networking	Y
	Improved contextualization	N
	Support for multiple storage back-ends	N
	Support for additional VLAN models	N
	Automatic setup of firewall rules	N
Service Manager	Claudia (Service Lifecycle Manager)	Y
	Claudia (Optimization Module)	Y
	Load Balancer Support	N
	Virtual Machine scalability driven by KPIs	N
	IP management	N
Monitoring	Physical and virtual infrastructure monitoring	Y
	Feeding OpenNebula with Ganglia information	Y
	Feeding Claudia with Ganglia information	Y
	Integration with grid service monitoring	N
Accounting	Grid site accounting integration with OpenNebula	N
	Grid-service accounting information	N
	Providing recommendations to grid accounting services	N

3 Implemented Features

This section provides an overview of the implemented functionalities in StratusLab with respects to automatic deployment and dynamic provision of grid services, as well as scalable cloud-like management of grid site resources. Table 3.1 presents an overview of the developed functionalities. This table lists the functionality, the repository where the software is located, the RPM packages which contain the functionality, whether the functionality is part of the StratusLab 1.0 release, administrator documentation, user documentation, and the contribution from within WP6. The table summarizes the implemented features which are described in the following sections.

3.1 Cloud like APIs

StratusLab works towards the use of cloud-like Application Programming Interfaces (APIs) for managing cloud computing capabilities including resource sharing [4]. That is, service providers (e.g. grid users) can use programmatic APIs to access to the shared resources in order to manage them and deploy their grid services. For this, StratusLab provides two implemented APIs for accessing to the service manager and the virtual machine manager: TCloud and OCCI.

3.1.1 TCloud API

The TCloud API [5] is a RESTful API which allows for accessing both the service manager and the virtual machine manager. The TCloud API has been implemented and released by the `tcloud-server` project, which is a general TCloud API representation and which is bound to an implementation by a set of drivers, for instance, the OpenNebula driver. The tcloud-server plus the drivers are provided as an RPM package `tcloud-server-rpm`, which is configured and installed by using the StratusLab sysadmin tools with the –claudia option.[1]

3.1.2 OCCI API

The OpenNebula OCCI API is a RESTful service to create, control and monitor cloud resources based on the OGF OCCI 1.0 API specification [2]. This feature is not included in StratusLab 1.0.

[1] See http://stratuslab.eu/doku.php/claudia.

3.1.3 Authentication and authorization in OpenNebula

The authentication and authorization are separated into two services. An authentication proxy identifies users and passes the user information to the virtual machine manager. OpenNebula, through its "auth" framework and a new "auth" module, will then make authorization decisions based on the specific request. OpenNebula will create new users dynamically, assigning them the default rights and quotas. Multiple mechanisms can be used simultaneously: username/password pairs maintained in a configuration file, username/password pairs from an LDAP server, grid certificates, and VOMS proxies created from grid certificates. The authentication proxy was developed in WP2, with support from WP6. The "auth" module was evolved in WP4, from an existing one developed in WP6. The OpenNebula "auth" framework was developed in WP6.

3.2 Virtual Appliance Language Definition

3.2.1 Open Virtualization Format

The Open Virtualization Format (OVF) [1] defines a portable packaging mechanism to foster the adoption of Virtual Appliances (VApp) as a new software release and management model (through the development of virtual appliance lifecycle management tools) in a vendor and platform neutral way. StratusLab provides a library to process OVF files as part of the Claudia distribution software. The `ovf-manager` is the OVF processing library which is part of the Claudia distribution.

3.3 Virtual Machine Manager

OpenNebula is at the heart of the StratusLab system, providing VM management. OpenNebula is a fully open-source toolkit to build IaaS private, public and hybrid clouds.

3.3.1 Improved fault tolerance

Fault tolerance has been improved in OpenNebula 2.2 to automatically trigger recovery actions when a physical host or VM fails.

3.3.2 Cluster support

OpenNebula 2.0 introduced support to cluster physical hosts. By default, all hosts belong to the "default" cluster. The administrator can create and delete clusters, and add and remove hosts from these clusters using the onecluster command. Thanks to this feature, the administrator can logically group hosts by any attribute like the service provided (e.g. CLUSTER = production), the physical location (e.g. CLUSTER = roomA) or a given characteristic (e.g. CLUSTER = intel).

3.3.3 User interface (Sunstone)

OpenNebula Sunstone is the new OpenNebula Cloud Operations Center, a GUI intended for users and administrators, that simplifies the typical management operations in private and hybrid cloud infrastructures. Virtual and physical resources can be managed in a similar way as it is done with the CLI.

3.3.4 Improved networking

Virtual networks of “fixed” type have been improved to allow its dynamic modification without the need of re-create them.

3.4 Service Manager

StratusLab introduces a layer on top of current IaaS clouds that provides users with the required abstraction level to allow them to manage the service as a single entity. This solution provides a wider range of scalability mechanisms and a broader set of actions that can be undertaken (addition, removal, reconfiguration, federation, etc.) than with a pure IaaS cloud.

The Claudia platform [6] is an advanced service management toolkit that allows service providers to dynamically control the service provisioning and scalability in an IaaS cloud. Claudia manages services as a whole, controlling the configuration of multiple VM components, virtual networks, and storage support by optimizing the use of them and by dynamically scaling up/down services by applying elasticity rules, SLAs and business rules.

3.4.1 Service Lifecycle Manager

Claudia is included in the StratusLab release for managing the service as a whole, controlling the configuration of multiple VM components, virtual networks, and virtual machine storage support like Network Attached Storage (NAS) and Storage area network (SAN).

3.4.2 Service Scalability

The Claudia platform supports the service optimization by dynamically scaling up/down services by applying elasticity rules, SLAs and business rules.

3.5 Monitoring

3.5.1 Monitoring the Physical and Virtual Infrastructures with Ganglia

Ganglia has been installed and tested in order to monitor the physical infrastructure hosting the reference cloud services. This monitoring system can be easily extended to monitor other aspects of the host fabric such as the number of VMs running in a specific host. Towards this, a couple of Ganglia probes have been developed that integrate with the underlying OpenNebula VM manager.

3.5.2 Feeding OpenNebula with Ganglia Information

Once ganglia is deployed in the infrastructure, new information drivers can be used to get information about hosts and virtual machines from it. These drivers should make the monitoring more efficient and more scalable in large installations as they do not use SSH connections to the nodes to get the information.

3.5.3 Feeding Claudia with Monitoring Information

OpenNebula provides Virtual Machine hardware monitoring information obtained from Ganglia. Claudia needs to obtain this virtual hardware information to scale grid services. Thus, this monitoring information collected by OpenNebula has to be provided to the Claudia optimization module.

Table 3.1: *Implemented Features Summary*

Topic	Repository	Packages	In 1.0?	Admin Docs	User Docs	WP6 work
TCloud API	http://code.stratuslab.eu/public/git/stratuslab-claudia.git	tcloud-server-rpm	Yes	http://stratuslab.eu/doku.php/claudia	http://claudia.morfeo-project.org/wiki/index.php/TCloud_Server	evolved existing code
OCCI API	git://git.opennebula.org/one.git	one-2.2-StratusLab	No	http://opennebula.org/documentation:rel2.2:occicg	http://opennebula.org/documentation:rel2.2:occiug	evolved existing code
Authentication and authorization in OpenNebula	http://code.stratuslab.eu/public/git/stratuslab-authn.git	stratuslab-cloud-proxy and one-2.2-StratusLab	Yes	http://www.stratuslab.eu/doku.php/authentication	Not applicable	The OpenNebula “auth” framework
OVF as the VApp language (ovf-manager library)	http://code.stratuslab.eu/public/git/stratuslab-claudia.git	clotho-rpm	Yes	http://claudia.morfeo-project.org/wiki/index.php/OVF_Manager	http://claudia.morfeo-project.org/wiki/index.php/OVF	Taken from Claudia
Improved fault tolerance	git://git.opennebula.org/one.git	one-2.2-StratusLab	Yes	Not applicable	http://opennebula.org/documentation:rel2.2:ftguide	Developed in WP6
Cluster support	git://git.opennebula.org/one.git	one-2.2-StratusLab	Yes	Not applicable	http://opennebula.org/documentation:rel2.2:cluster_guide	Developed in WP6

Graphical User interface	git://git.opennebula.org/one.git	one-2.2-StratusLab	No	http://opennebula.org/documentation:rel2.2:sunstone#requirements_installation	http://opennebula.org/documentation:rel2.2:sunstone#usage	Developed in WP6
Improved networking	git://git.opennebula.org/one.git	one-2.2-StratusLab	Yes	Not applicable	http://opennebula.org/documentation:rel2.2:vgg	Developed in WP6
Claudia (Service Lifecycle Manager)	http://code.stratuslab.eu/public/git/stratuslab-claudia.git	clotho-rpm, tcloud-server-rpm, claudia-client-rpm	Yes	http://stratuslab.eu/doku.php/claudia	http://stratuslab.eu/doku.php/claudia#claudia_stratuslab_test	Developed in WP6
Claudia (Optimization Module)	http://code.stratuslab.eu/public/git/stratuslab-claudia.git	clotho-rpm, tcloud-server-rpm, claudia-client-rpm	Yes	http://stratuslab.eu/doku.php/claudia	http://stratuslab.eu/doku.php/claudia#claudia_stratuslab_test	Developed in WP6
Physical and virtual infrastructure monitoring	git://git.opennebula.org/one.git	one-2.2-StratusLab	Yes	http://stratuslab.eu/doku.php/ganglia	Not applicable	Probes developed in WP6.
Feeding OpenNebula with Ganglia information	git://git.opennebula.org/one.git)	one-2.2-StratusLab	Yes	http://opennebula.org/documentation:rel2.2:ganglia	Not applicable	Developed in WP6.

Integration with grid service monitoring	http://code.stratuslab.eu/public/git/stratuslab-claudia.git	reportclient-rpm	Yes	Not applicable	Not applicable	Developed in WP6.

4 Missing Features and Next Steps

This section provides an overview of the missing features in the current release and next steps for development.

4.1 Cloud like APIs

- Integrate TCloud with the authentication and authorization proxy: All the Claudia requests will be done by the authentication and authorization proxy so that unauthorized users will not be able to perform service management requests.
- Extension of authorization to support of groups of users and access rules (roles) to manage OpenNebula resources. These groups and roles are needed to map VOMS attributes.
- `tcloud-server` to be used as a separated component. Currently, the tcloud-server and drivers code are part of the claudia distribution, and are installed with it. The idea is to separate the tcloud-server package to be installed and configured independently from claudia for its usage like the OpenNebula API.

4.2 Virtual Machine Manager

- Improvements in contextualization by incorporating additional attributes like complete resource descriptions.
- Support for multiple storage backends to access the image repository (for persistent images).
- Support for additional VLAN models for virtual network management.
- Implementation of the automatic setup of simple TCP firewall rules for virtual machines.

4.3 Service Manager

- Load Support in Claudia. Elasticity constraints require a service that can balance the load between the different replicas. Claudia should provide a support for providing the replica information required by the load balancer.

- Virtual Machine scalability driven by KPIs.
- Removing the IP management from Claudia and relying on the OpenNebula management.

4.4 Accounting

- Integration of grid site accounting with OpenNebula accounting. Two grid accounting systems will be evaluated: APEL (Account Processor for Event Logs), which is the most popular system and currently installed in the production site running over the StratusLab reference cloud services , and DGAS (Distributed Grid Accounting System).
- Expand the existing OpenNebula capabilities to accommodate grid-service related accounting information (e.g. VO-related information).
- Provide recommendations for cloud-friendly expansions to existing grid accounting services. These expansions will depend on the resource accounting model chosen by the project. These expansions will have to be further negotiated with EMI. This negotiation will take place in the context of the activities defined in the MoU signed between StratusLab and EMI (currently in progress).

4.5 Grid Site Elasticity

- Enable grid-site installation and configuration through Claudia. A set of scripts have to be developed that parse the grid-site definition from the OVF file and generate the necessary YAIM configuration files. In the last step these scripts will also trigger the YAIM execution in order to complete the site configuration.
- Enable grid-site elasticity. The site capacity (number of WNs) should be able to adapt to current processing demands. Elasticity management will be done through Claudia by defining the necessary KPIs. This will also require the cooperation of Claudia with the underlying LRMS (e.g. Torque) in order to trigger the site expansions or contraction depending on a set of rules (e.g. Increase site capacity by 20% if site utilization surpasses 80%). For the latter we will have to develop the necessary logic that will reside on the site Computing Element.

5 Conclusions and Future Work

This document has provided an overview of the functionalities implemented within the scope of WP6 in the first period of the project, which are intended to close the identified gaps in D4.1. Thus, D6.1 defined a set of solutions and D6.2 has described the status of implementing them.

These developments provide a RESTful Cloud API for accessing to the service manager and the virtual machine manager (TCloud and OCCI) with authentication and authorization functionalities. Moreover, it has provided a processing library to support the Virtual Appliance Language for the services to be deployed including the deployment of a grid site. Regarding service management, an open source framework (Claudia) has been included to control the service and provide scalability support according to user-defined rules. In addition, some improvements have been made in the Virtual Machine Manager for improving fault tolerance, networking, and cluster support. In monitoring, the physical and virtual machine infrastructure monitoring information have been provided by Ganglia and used by OpenNebula and Claudia.

However, there are still some functionalities which have not been implemented yet and are in the roadmap to be developed in the near future. In the short term, grid site deployment and elasticity are going to be done automatically by using Claudia. Claudia is going to provide grid site scalability driven by KPIs and provide load balancing support which can be use to manage the Worker Nodes deployed in the grid site.

Equally important is the issue of resource accounting and the integration of grid sites accounting system with the respective cloud services accounting. This issue has been identified from the early stages of the project but work on it had to be postponed until we gained enough experience from the operation of cloud and grid services in the context of WP5. This experience will help us identify an integration path between the two worlds and design a solution for grid/cloud interoperation. In this process it will be also necessary to include relevant stakeholders from the domain of grid operations (e.g. through the collaboration with EGI-InSPIRE project) and middleware development (EMI).

Finally, more changes in Claudia and OpenNebula are going to be developed in order to satisfy StratusLab requirements.

Glossary

APEL	Account Processor for Event Logs
Appliance	Virtual machine containing preconfigured software or services
Appliance Repository	Repository of existing appliances
CDDLM	Configuration Description, Deployment, and Lifecycle Management
DGAS	Distributed Grid Accounting System
DHCP	Dynamic Host Configuration Protocol
DMTF	Distributed Management Task Force
Front-End	OpenNebula server machine, which hosts the VM manager
Hybrid Cloud	Cloud infrastructure that federates resources between organizations
IaaS	Infrastructure as a Service
IP	Infrastructure Provider
Instance	a deployed Virtual Machine
JRA	Joint Research Activity
KPI	Key Performance Indicator
Machine Image	Virtual machine file and metadata providing the source for Virtual Images or Instances
NFS	Network File System
Node	Physical host on which VMs are instantiated
OASIS	Organization for the Advancement of Structured Information Standards
OCCI	Open Cloud Computing Initiative
OGF	Open Grid Forum
OVF	Open Virtualization Format
Public Cloud	Cloud infrastructure accessible to people outside of the provider's organization
Private Cloud	Cloud infrastructure accessible only to the provider's users
Regression	Features previously working which breaks in a new release of the software containing this feature
Service Manager/SM	A toolkit to provides Service Providers to dynamically control the Service provisioning and scalability
Service Provider/SP	The provider who offers the application to be deploy in the Cloud
SMI	Service Manager Interface
SSD	Solution Deployment Descriptor

VApp	Virtual Appliance: pre-configured software stacks comprising one or more virtual machines to provide self-contained services
Virtual Machine / VM	Running and virtualized operating system
VMI	Virtual Manager Interface
VO	Virtual Organization
VOMS	Virtual Organization Membership Service
Web Monitor	Web application providing basic monitoring of a single StratusLab installation
Worker Node	Grid node on which jobs are executed

References

[1] DMTF. Open virtualization format specification. Specification DSP0243 v1.0.0d. Technical report, Distributed Management Task Force, Sep 2008. https://www.coin-or.org/OS/publications/optimizationServicesFramework2008.pdf.

[2] OCCI-WG. Open Cloud Computing Interface Specification. Technical report, Open Grid Forum, 2009. http://forge.ogf.org/sf/go/doc15731.

[3] Stratuslab Consortium. Deliverable 4.1 Reference Architecture for StratusLab Toolkit 1.0. Online resource., 2010. http://stratuslab.eu/lib/exe/fetch.php?media=documents:stratuslab-d4.1-v1.0.pdf.

[4] Stratuslab Consortium. Deliverable 6.1 Cloud-like Management of Grid Sites 1.0 Design Report. Online resource., 2010. http://stratuslab.eu/lib/exe/fetch.php/documents:stratuslab-d6.1-v1.0.pdf.

[5] Telefónica. TCloud API Specification, Version 0.9.0. Online resource., 2010. http://www.tid.es/files/doc/apis/TCloud_API_Spec_v0.9.pdf.

[6] TID. The Claudia project. Online resource., 2010. http://claudia.morfeo-project.org/wiki/index.php/Main_Page.